BOLT BERANEK AND NEWMAN INC

CONSULTING · DEVELOPMENT · RESEARCH

Job No. 11126

STUDIES OF MANUAL CONTROL SYSTEMS
Progress Report No. 4

for the Period 19 April 1964 to 18 July 1964

1 October 1964

Submitted to:

National Aeronautics and Space Administration
600 Independence Avenue
Washington, D. C. 20546

Attention: Dr. T. L. K. Smull

STUDIES OF MANUAL CONTROL SYSTEMS

PROGRESS REPORT NO. 4

for the Period 19 April 1964 to 18 July 1964

Contract NASw-668

1 October 1964

Submitted to:

National Aeronautics and Space Administration
600 Independence Avenue
Washington, D. C. 20546

Attention: Dr. T. L. K. Smull

TABLE OF CONTENTS

STUDIES OF MANUAL CONTROL SYSTEMS

I. INTRODUCTION

This report is for the first quarter of the second year of work on Contract NASw-668. During this second year of the contract we are focusing on two principal tasks: (1) investigation of computer facilities for analysis and modelling of the results of tracking experiments; and, (2) investigations of the multivariable control characteristics of the human operator.

During this first quarter we have concentrated on the first of these two tasks. We have devoted almost all of our effort to the design and programming of a system which we call Signal Analyzer I. It is a digital computer system for analyzing the signals and identifying the dynamic characteristics of the elements of manual control systems. Signal Analyzer I is an improved, augmented version of the multiple regression analysis system that we have been using for several years, which was developed initially under Contract NASw-185 and Contract AF33(657)10124.

This progress report will be devoted to a description and discussion of Signal Analyzer I. First we present an operational description of the system, that is, a description of the system from the point of view of the user. Then we will discuss in some detail the organization of the system and some of the principal programs that constitute it.

II. OPERATION OF SIGNAL ANALYZER I

Signal Analyzer I is a system of programs written for the BBN PDP-1 computer. The key task of Signal Analyzer I is to determine the describing functions of the human operator and other elements of a control system, according to the multiple linear regression analysis paradigm developed by Elkind et al.[1] In so doing, it can also determine the characteristics of the signals flowing in such a control system and provide time or frequency displays of them in much the same way as an oscilloscope or a spectrum analyzer would.

The system is built around the basic philosophy of providing close communication between the computer and the experimenter. Thus, the experimenter is given many opportunities to enter the system to check partial results, change the parameters of the analysis, make decisions, examine the results in various ways, and set up the next run of the analysis process using the modifications suggested by his previous interaction with the system.

Communication is handled by means of typeouts in the form of questions to be answered by the experimenter, or, in more complicated situations, by means of visual displays, which enable decisions to be made by pointing an arrow to the desired command and introducing the desired changes by means of the typewriter.

In the following sections, a brief summary of the modelling technique used is given first, followed by a detailed description of the system from the point of view of the operator.

A. MODELLING TECHNIQUE

Consider the control system as represented in Fig. 1. It is desired to model or describe the human operator's transfer characteristics. The multiple regression analysis paradigm represented in Fig. 2 will be used to obtain this description.

In this model, the human pilot's impulse response is approximated by a finite series of orthonormal functions, ϕ, whose coefficients, b, are determined so that the output of the model, z, differs the least from the human pilot's response, y, in a mean square error sense. As it can be readily found, the coefficients b (which are regression coefficients), are given by the matrix equation

$$\underline{L}\,\underline{b} = \underline{y}$$

where $\underline{L}$ is the covariance matrix of the filter outputs, z_i, and $\underline{y}$ is the covariance vector of the human pilot's response with the filter outputs. The elements of $\underline{L}$ and $\underline{y}$ are usually computed by taking the time averages of $z_i z_j$ and $z_i y$, respectively.

The orthonormal functions used are the exponentials described by Huggins.[2] While in the present version of the system only real axis poles are allowed, they nevertheless provide very good approximations to characteristics of the human pilot.

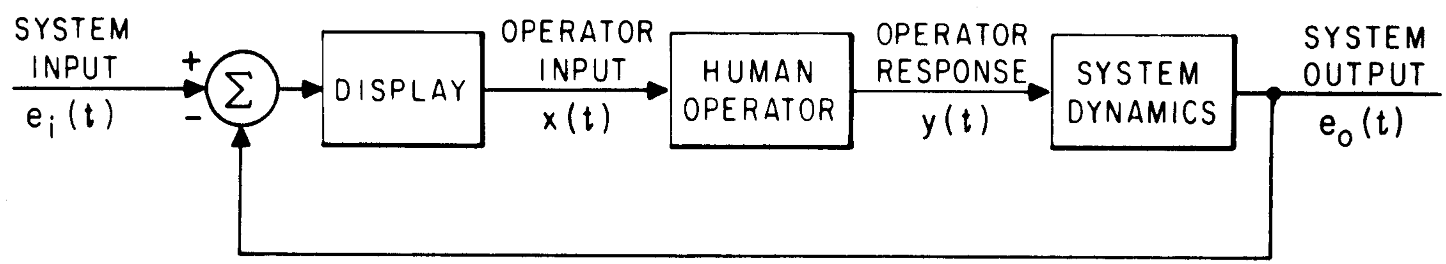

Figure I Block Diagram Of A Simple Manual Control System

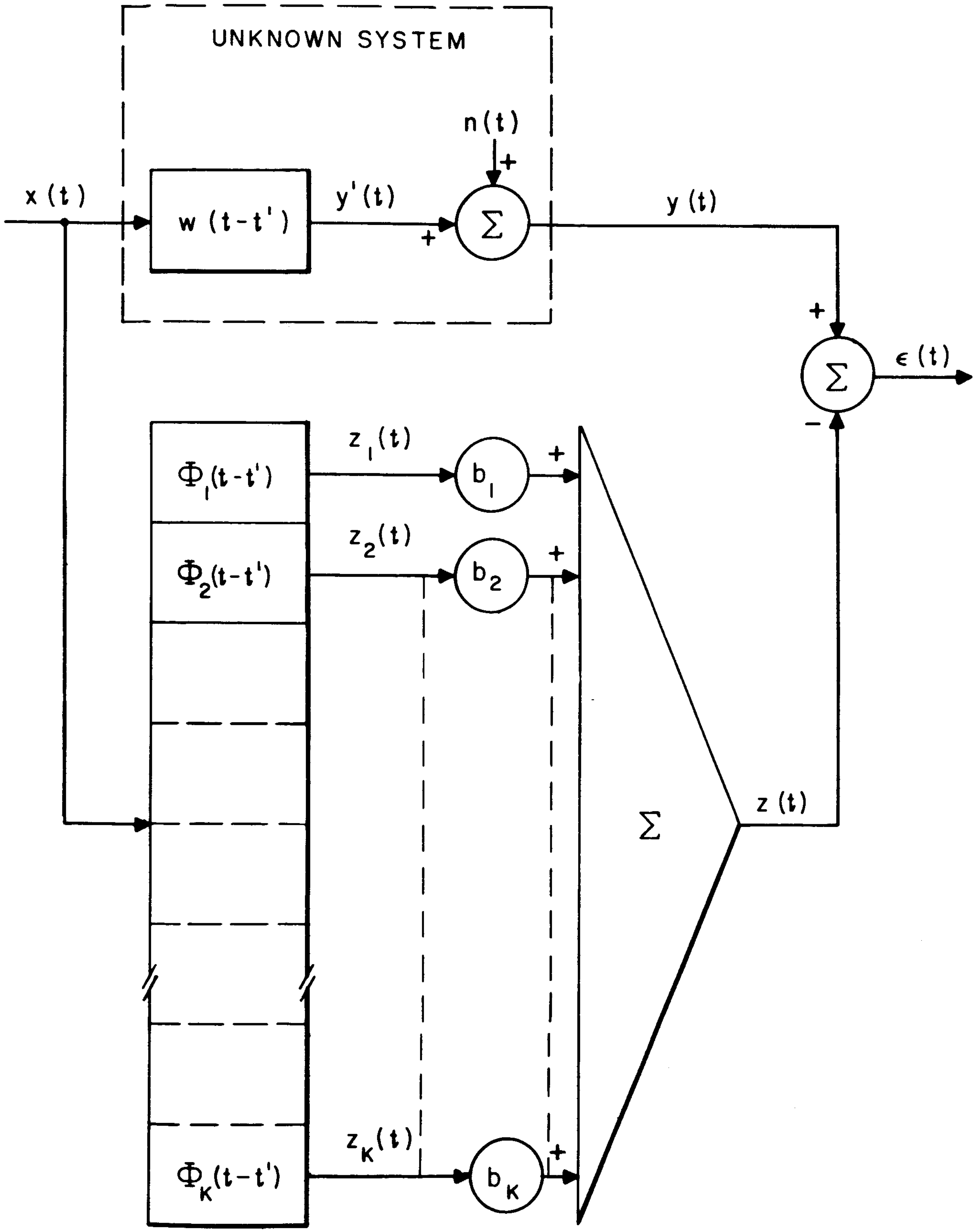

Figure 2 Multiple Regression Analysis Paradigm For Identification Of Dynamic Systems

The power spectra of signals can be determined using this same basic technique. A complete description of the method is given by Elkind et al.[1, 3]

B. OPERATIONAL DESCRIPTION

In this section we attempt to convey to the reader the impression that he is actually the experimenter conducting a simulation study of a human pilot controlling a space vehicle. We take you -- the reader -- on a "simulated visit" to the computer room at BBN, where we give you a demonstration of Signal Analyzer I and explain how it is used to analyze the results of an experiment.

You have been here, in the computer room, for about one-half hour already and you have witnessed how the data that you are going to analyze next have been generated, recorded, and edited. You have seen a human pilot engaged in a control task, namely: controlling the attitude of a space vehicle; you have watched, on a multi-channel display, the signals generated in the experiment while they were being sampled and stored in real time in the digital computer's memory. Later you saw how the signals associated to a particular event that happened to be of interest were sliced off from the rest, labelled, indexed, and stored on magnetic tape for future analysis.

Now the time has come to perform the analysis.

The event you are interested in studying is the change in the observed human pilot describing functions after the controlled-element dynamics have undergone a change. More precisely, the space vehicle control dynamics were suddenly changed, at time $t_o + 8$, form k/s^2 to $-2k/s^2$. Consequently, you have selected for analysis the portion of the signals recorded during the period t_o to $t_o + 25.6$ secs.

At this point you sit in front of both the typewriter and the 'scope. The analysis portion of Signal Analyzer I is started. The first thing you see is a display of a control panel, like the one shown in Fig. 3. The purpose of this display is to present the fundamental parameters of the analysis to be performed and to allow you to change these parameters if you want to do so.

```
DATA

  block on tape          20
  h (sec)                +1.0000-01

BODE PLOT

  WMIN (rad/sec)         +6.2499-02
  octaves (nmbr of)      8
  N (frqncy pnts)        64

ANALYSIS

  nsmax (smpl pnts)      50
  nO (intl anlys pnt)    30
  amax (max dly pnt)     1

S, FILTER POLES

  K (nmbr fltrs)         5
            -1.0000+00
            -2.0000+00
            -3.9999+00
            -8.0000+00
            -1.6000+01
```

Figure 3

For example, to specify the set of signals to be analyzed you move, by typing u (for up) or d (for down), the little arrow (appearing near the bottom of Fig. 3) until it points to "block on tape." Then you type in the number of the block containing the desired signals.

Once you are satisfied with your choice of parameters, you flip a sense switch and the program continues and reads the specified block from magnetic tape.

When this is done, the label and indexing information of the block is typed out and the signals corresponding to error, e(t), and pilot output, y(t), in Fig. 1 are displayed as functions of time as in Figs. 4 and 5.

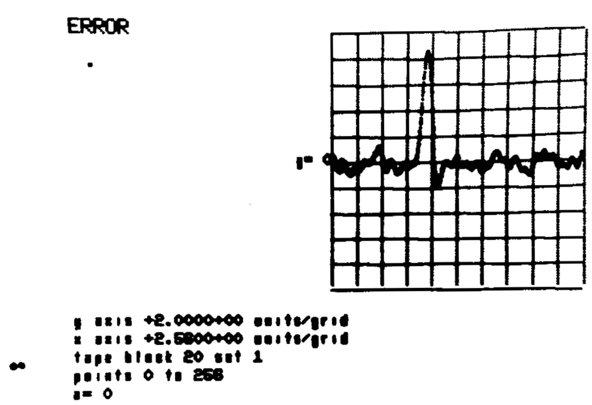

Figure 4

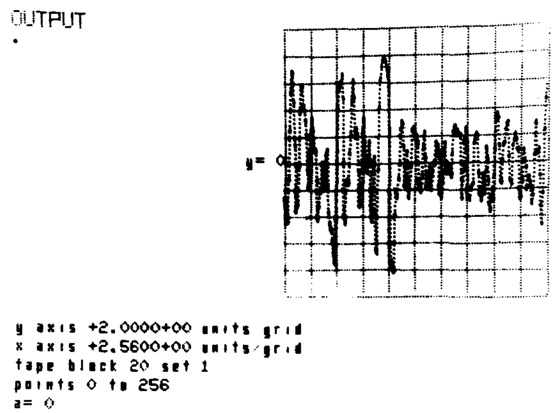

Figure 5

While looking at Fig. 4 you notice how the error increases greatly right after the change in the controlled-element dynamics. The large stick movements the human pilot made while he was trying to cope with the new controlled-element dynamics are evident in the output signal of Fig. 5. Having seen the signals that are going to be analyzed, you may wish to go back and reset some of the parameters that you tentatively assigned while you were looking at the display of Fig. 3. For example, you may wish to reset the initial point, n0, and the number of datum points, nsmax, (the length of the segment) to be analyzed so that you will have a description of the human pilot's characteristics right before the control change. To do so you flip a switch (the same for all displays) and you answer NO (by typing) when the computer asks OK?. This puts you right back where you wanted to be -- the set up display is on the oscilloscope --

and you make the desired changes. This pattern is repeated as long as necessary until you are satisfied, a variety of questions being asked and answered in the same way during the intervening process.

Once you are all set and answer <u>YES</u> to that final <u>OK?</u>, the computation begins and the filters you called for by specifying their poles are "implemented" (their impulse responses and transfer functions are calculated).

After this, the computer asks <u>LOOK FILTERS?</u>, which means: do you want to see if the poles you specified for the analysis filters produce the frequency or time domain characteristics you thought they would?. If your answer is <u>YES</u>, further questioning leads you to specify what characteristics (impulse response or frequency response) and which filter you wish to see.

Signals are displayed in the time domain by a program called <u>Time Display</u> while transfer functions are displayed in the frequency domain by a program called <u>Frequency Display</u>. <u>Time Display</u> acts essentially as an oscilloscope, allowing changes in sweep rate, vertical gain and number and size of grid divisions. These changes are effected by flipping another sense switch which causes the appearance of a "control panel" with which the changes are made. Let us suppose you wanted to look at the impulse response of filter 3. This is what you would see:

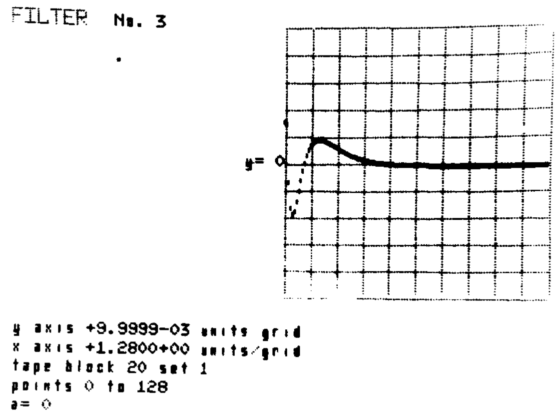

Figure 6

The control panel for this display (the "oscilloscope" setting) looks like this:

N, grid divisions	10
→ W, graph size	10
T, units/grid	+1.2800+00
y, units/grid	+9.9999-03
ygain, units/volt	+9.9999-01

Figure 7

<u>Frequency Display</u> acts either as a Bode plotter, when amplitude and phase information are to be presented, or as a spectrum

plotter, when only magnitude is to be presented. Let us suppose you wanted to see the Bode plot of the transfer function of filter 3 displayed on a semi-log graph with angular frequency on the abscissa and decibel amplitude ratio and phase angle in degrees on the ordinate. This is what you see:

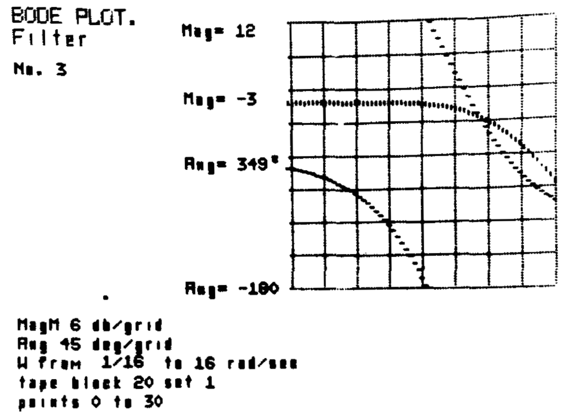

Figure 8

Here, the size of the grid divisions, as well as the origins of the ordinates, can be set up by the user by means of the "control panel" facility, but other parameters of the display are basic to the analysis to be carried out and were set forth in the display of Fig. 3. The "control panel" for this display is:

```
 W. graph size       10
 decibels/grid        6
 Mag. top grid       +1.1999+01
→Ang. bottom grid    -1.7999+02
```

Figure 9

Notice that the signal description, as well as other characteristics, appear on the display, but any additional information you may care to add can also be made to appear anywhere on the screen by means of the typewriter, as it was done with the additional "title" information in Fig. 10.

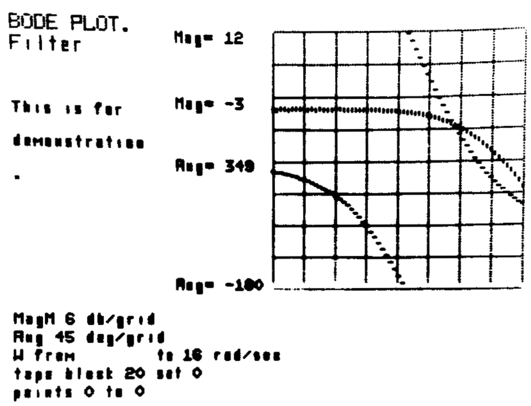

Figure 10

Once you are satisfied with your selection of filters, the analysis really begins. Upon specification of the initial state of the filters (they may be zeroed or used in the last state in which they were left), the regression coefficients are computed and typed out, along with the mean square residual (difference between model output and pilot output), the corresponding power in the pilot's output signal, and the estimated values of the standard deviations of the regression coefficients. This is repeated for each value of the delay up to the maximum value (amax in Fig. 3) that you specified. After this, you choose a delay according to your needs and communicate it to the machine.

This finishes the analysis phase and is followed by the presentation of the results, where you may choose any of the possibilities offered to you in the display of Fig. 11 by pointing (by typing u or d) the arrow to the desired result. Figs. 12 to 17 show examples of some of the possible choices offered. Results can be examined in any order and as many times as desired.

```
TIME DISPLAY
    Model Impulse Resp.
    Residual
    Mimic
    Error
    Output
    Z. which z

FREQUENCY DISPLAY
   Bode Plot

  POWER SPECTRA
    Residual
    Error
→   Output

ss 2 ^' [illegible] display chosen
     Exit altogether
```

Figure 11

For example, you may wish to look at the mimic output (the output of your model, as in Fig. 2) and compare it with the pilot's output. So you point the arrow to <u>Mimic</u>, flip the sense switch

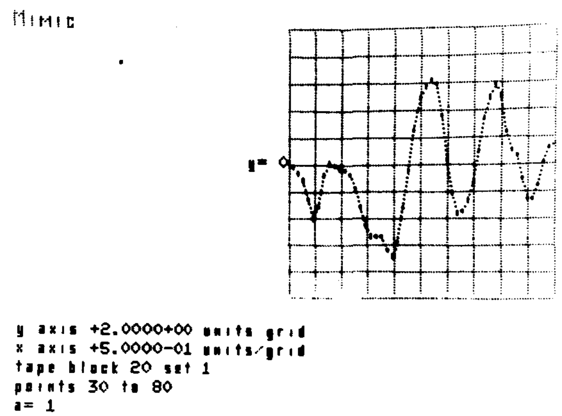

Figure 12

and then flip the sense switch, move the pointer to <u>Output</u>, and flip again.

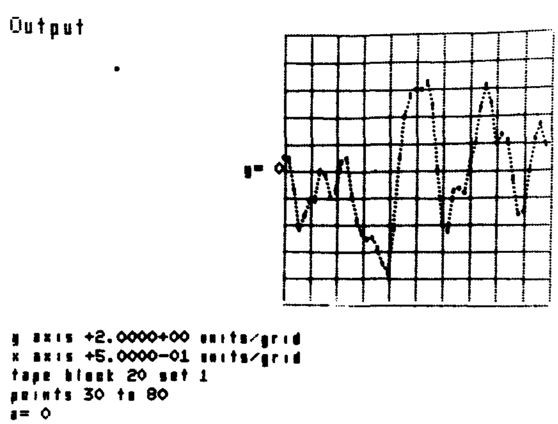

Figure 13

As you can see, the model seems to be a good one. Let us take a look at the residual

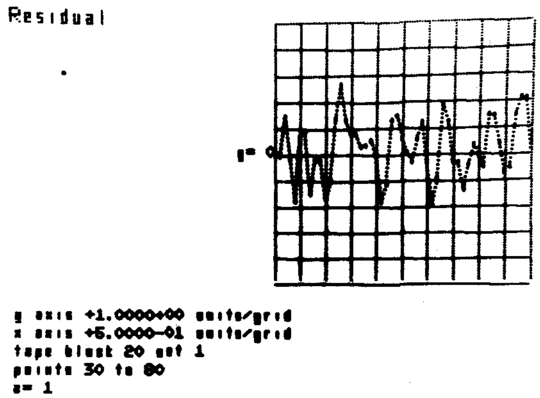

Figure 14

and its power spectra, as measured by the bank of filters you chose.

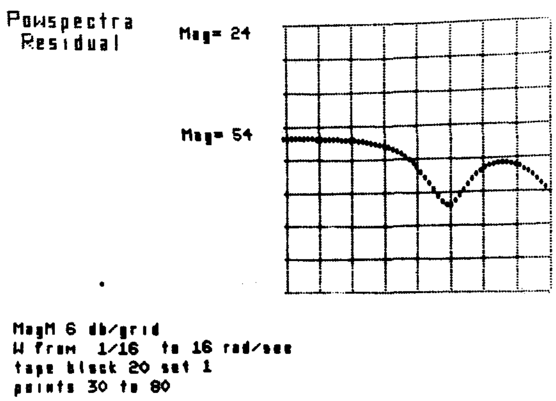

Figure 15

Finally, let us look at the Bode plot of the model, that is, the transfer function of the set of filters with their associated gains (regression coefficients).

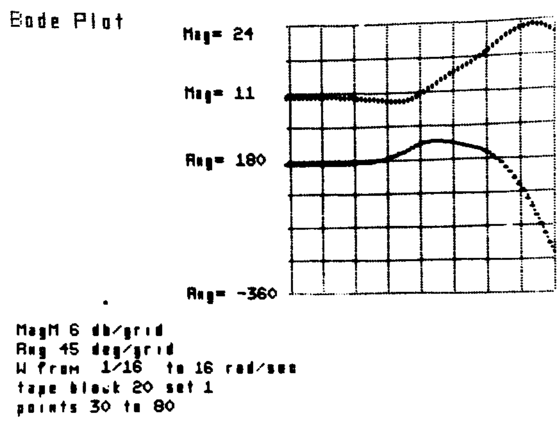

Figure 16

As you can see, the phase at low frequencies is nearly 180° while the magnitude level is 11 db. Remember that the controlled-element dynamics was of the form k/s^2 before the transition and $-2k/s^2$ right after the transition.

In order to continue with the analysis you leave the presentation stage by pointing the arrow to "Exit altogether" and specify (more questions are asked) the new segment to be analyzed, as well as other changes you may wish to incorporate

Let us suppose that you want to look at pilot's behavior 3 secs after the transition, that is, at t_o + 11 secs; and that the length of the segment is again 5 secs, or 50 datum points.

After the regression coefficients, mean square error, etc., are typed out and the delay chosen, you should be eager to look at the Bode plot of the model and see what has happened. Here it is:

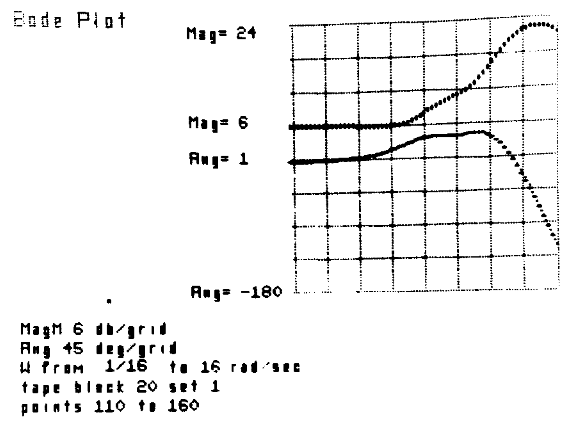

Figure 17

Indeed, the pilot has responded the way you would have expected. Look at the phase at low frequencies: it is now near 0 degrees as it should be to compensate for the polarity reversal of the controlled-element dynamics. And the magnitude has dropped too, although not quite the 6 db you would have predicted by the doubling of the gain in the controlled-element dynamics.

Probably the pilot has not yet had time to adapt himself completely to the new situation, in which case, subsequent plots should show a decrease in amplitude ratio.

Let us go on then and ....

At this point you decide that you have had enough for one day -- and so we conclude our "analog demonstration" of Signal Analyzer I.

III. SYSTEM ORGANIZATION

The Signal Analyzer I system was written in the Decal-BBN[4] programming language, which incorporates a number of ALGOL-like facilities (like Procedures, "for" statements, and "conditional" statements) and a powerful algebraic compiler that allows for subscripted variables and array handling as well as a fairly complete set of floating point subroutines.

The system's organization was largely dictated by the sizes and types of available memories. The BBN computer is a DEC-PDP-1[5] which has two 4,000 word core-memory modules with a cycle time of 5 microseconds, a magnetic drum with a total storage capacity of twenty-two 4,000 word core-loads and with an average random access time of 33 milliseconds, and two IBM compatible, magnetic tape units.

The system can be viewed as consisting of Programs and Data; consequently, a first allocation of memory space was made by assigning one core-memory module to each and as many drumfields as needed. This division was adopted mainly to simplify program writing. The data are organized in arrays of floating-point numbers and addressed by subscripted variables. Arrays are declared and exist solely on drumfields. When an array is needed, the whole drumfield containing the array is loaded into the data core.

Care must be exercised in programming computations, so as to avoid using arrays in different drumfields, since this could introduce too many drumfield swaps which are inherently slow (33 milliseconds).

The program structure is naturally more complicated than the data structure. Programs are organized in levels. At the first, or lowest level, are the fundamental programs like Filter, Impulse Response, Transfer Function, Covariance, Inversion, etc. They are written in the form of ALGOL Procedures, and the procedural call must contain the parameters needed by the program. For example, a procedure like Filter must be supplied with such information as: what signal is going to be filtered, where to put the filtered signal, the initial state of the filter, etc., which constitute what could be called the variables of the computation. Other parameters, such as the sampling interval, the location of the filter's poles, etc., which are constant throughout the analysis, are not communicated by the procedural call, but are kept in protected storage and are available to all the programs without regard to their level. As can be readily seen, this structure is highly modular, allowing incorporation of changes at the basic level without upsetting the rest of the system.

The second-level programs perform most of the logical functions of the system, handle the procedural calls, communicate with the experimenter, react to his decisions. A fundamental tool used by these second-level programs is a secretarial routine called Fetch. Fetch is used to communicate procedural calls across the drum. When a procedure is called which is not present in core, <u>Fetch</u> interprets the arguments of the call, brings in the drumfield containing the called procedure, hands back the arguments, transfers control to the procedure, and remembers where the call was originated. Upon completion of the procedure, which may encompass other "<u>Fetches</u>," the drumfield that was brought in is updated, the original drumfield is brought back and control is transferred to continue in sequence.

Finally, the third-level program, or <u>Master</u> routine, simply coordinates and calls the second-level programs. In Fig. 18 is a block diagram that shows the principal features of the <u>Master</u> Program. The program is entered and a display is presented that is used to select the signals to be analyzed and the parameters for the analysis such as the analysis filter poles, data to be analyzed, number of data points to be used, etc. A more detailed block diagram of the part of the system that performs this selection is in Fig. 19. The user is next given a choice of proceeding with the analysis or viewing the signals and analysis filter characteristics. In Fig. 20 is a diagram of the portion of the system that displays filter characteristics. The user next is given a choice of initializing the filters to rest state, or leaving them in their present state. After the initialization, the regression analysis is performed and a display is posted on the computer scope which contains a list of the results available from the program, for example, the impulse response and Bode plots of the system measured, and the power spectra of the input, output, and residual signals. In Fig. 21 is a block diagram of the portion of the system that computes and displays the desired results.

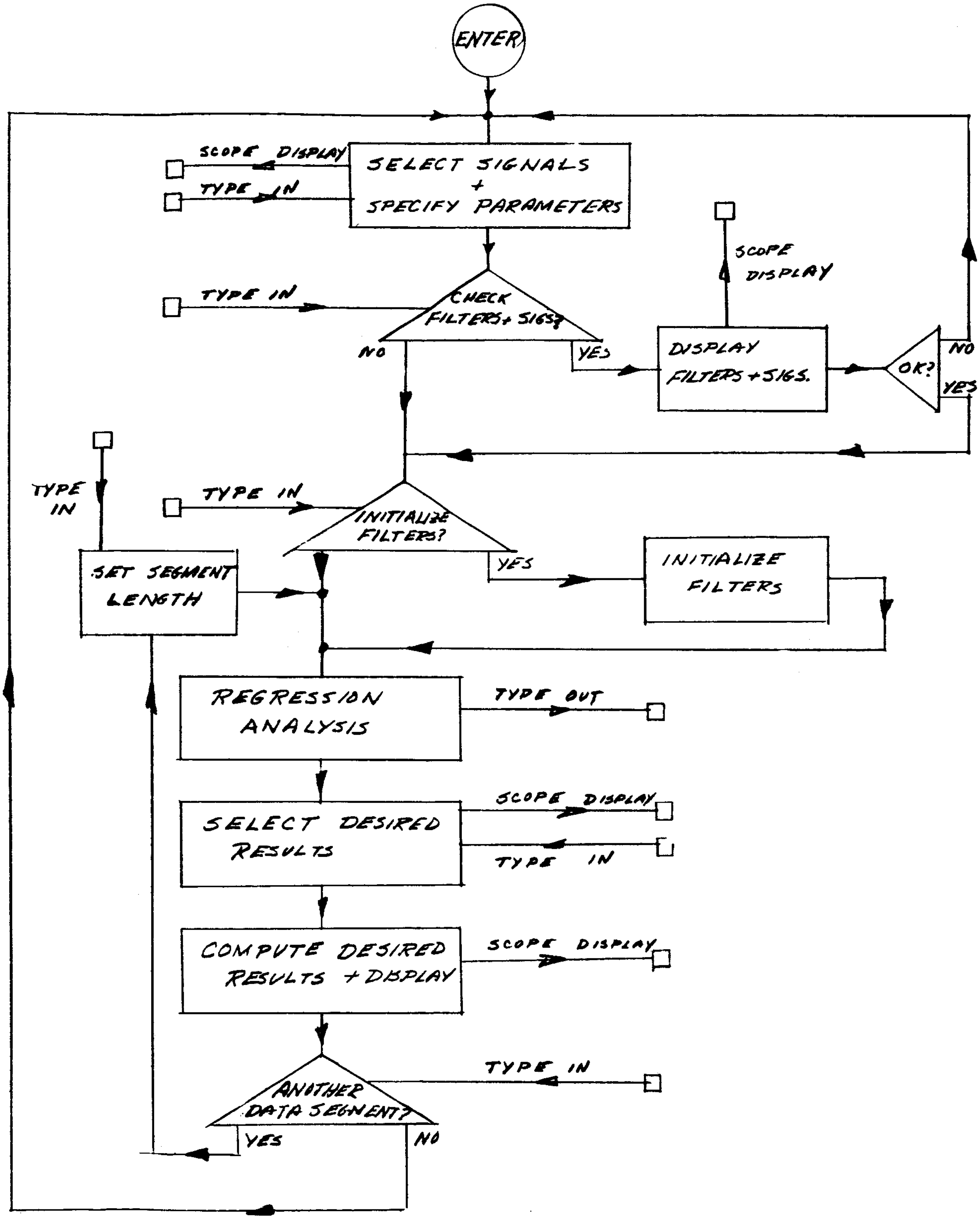

Fig. 18 Block Diagram of Principal Elements of Signal Analysis System.

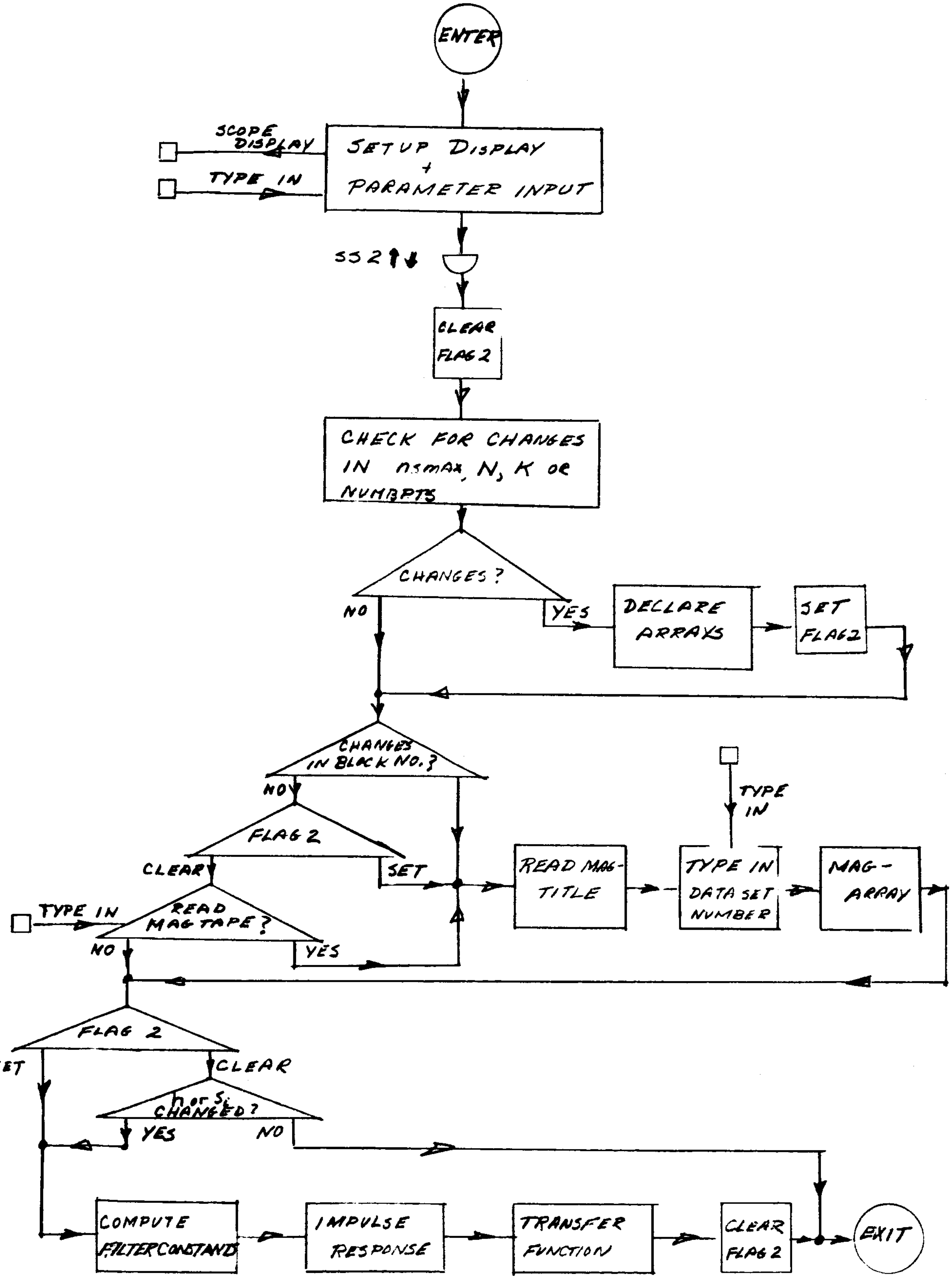

Fig. 19 Block Diagram of Signal Selection and Parameter Specification Part of System.

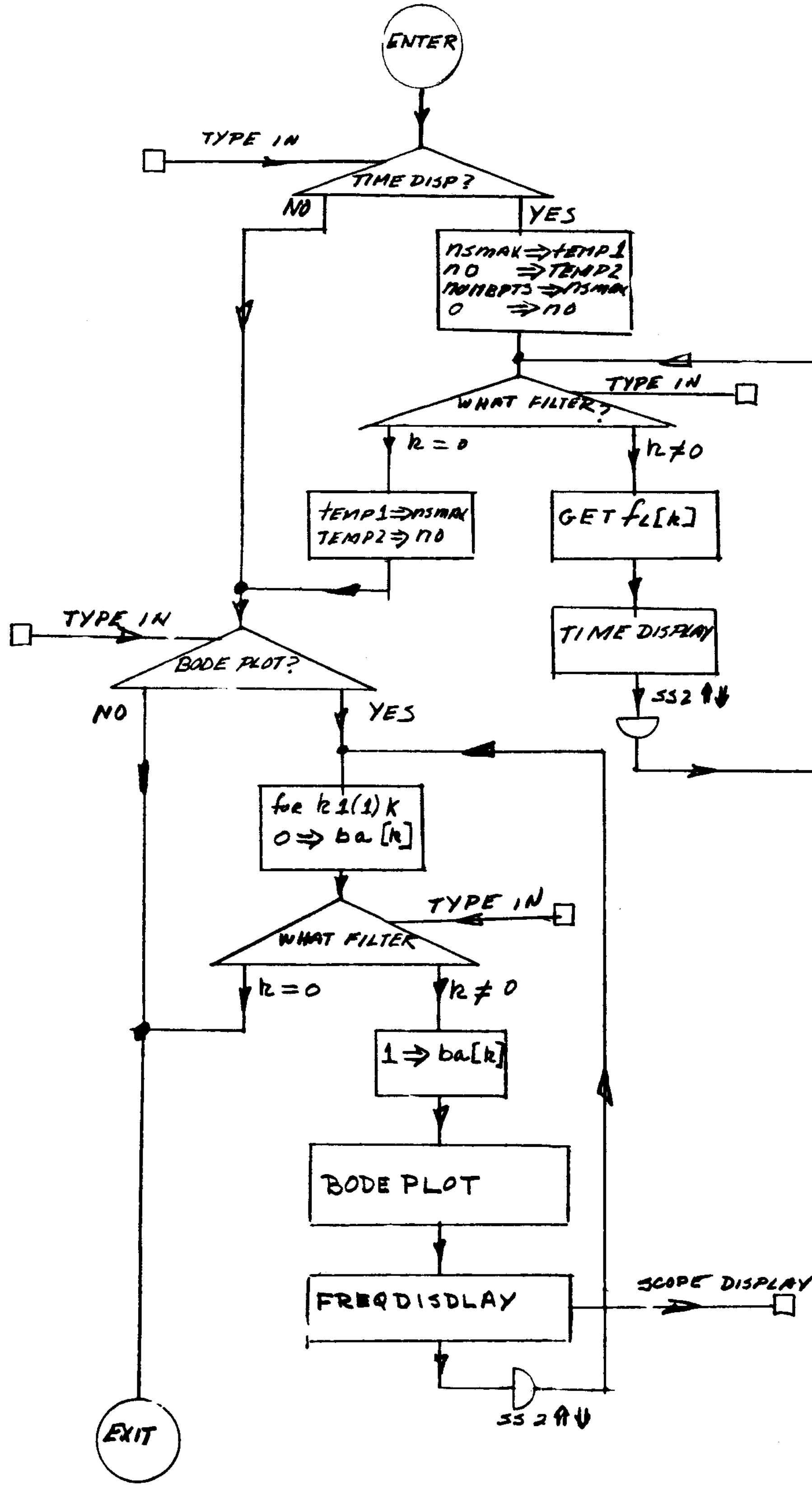

Fig. 20 Block Diagram of Filter Characteristics Display.

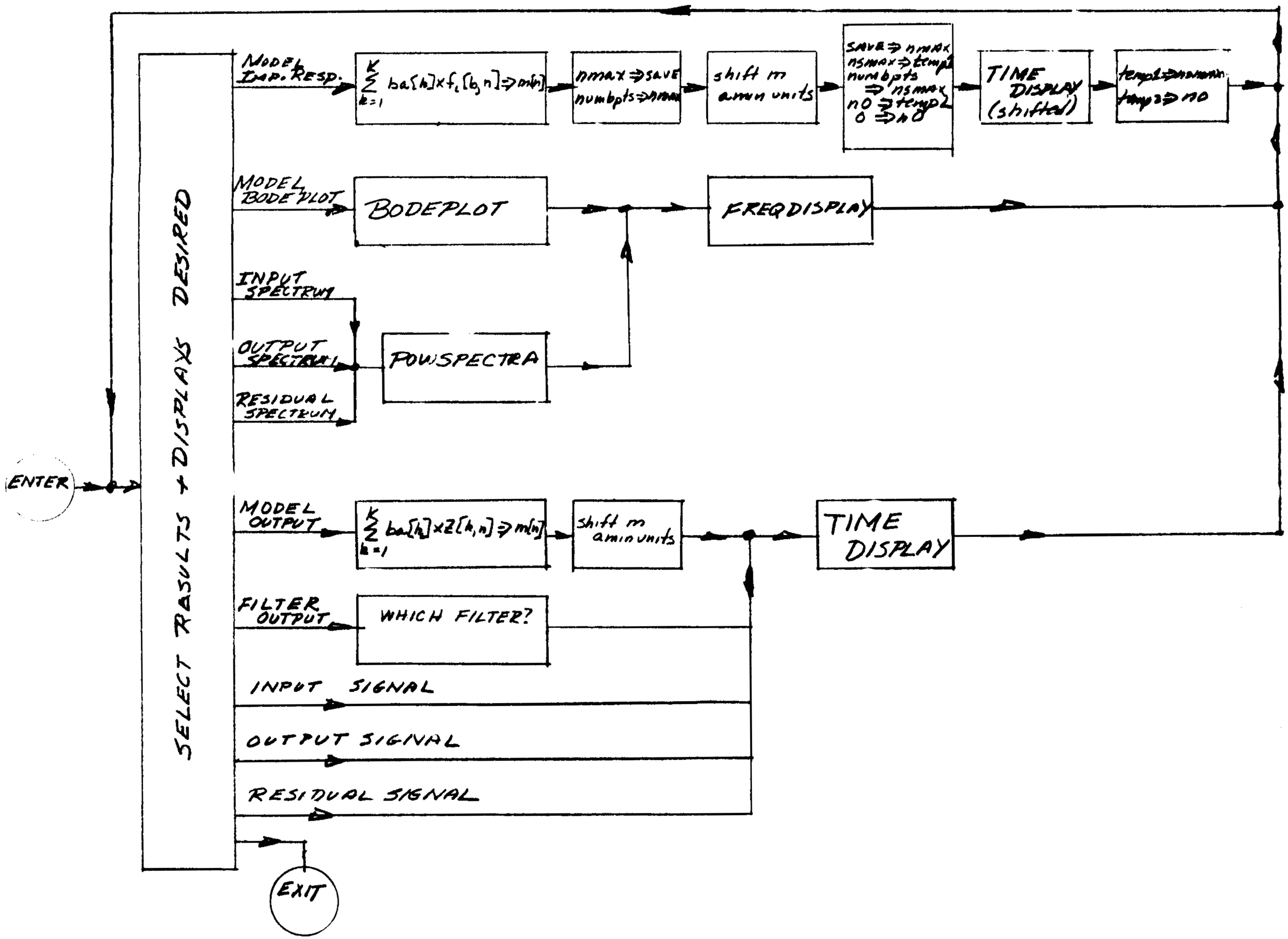

Fig. 21 Block Diagram for Selection, Computation, and Display of Desired Results.

IV. CONCLUSIONS

We have used Signal Analyzer I for the past few months in our studies of human controller characteristics. In some of these experiments the system was connected on-line to the experiment and was used to record the signals from the tracking experiment and to determine the characteristics of the controller and of the signals immediately after each experimental run. In other experiments the system was used to process data in semi-production runs. The on-line use of the system, in particular, proved to be very valuable. The immediate availability of experimental results led to considerable saving of experimental time largely because it permitted us to focus the experiment on the interesting conditions.

The display and control features of the system, which make man-computer communication easy and pleasant, have proven to be very worthwhile. Considerable effort went into the design of these features. The fact that the system is flexible and that its logical organization allows the user to take short cuts at every important decision node leads to reasonable efficiency in bulk processing of data.

Of course, as with any system of this kind, one sees a need for a number of improvements. These fall into two main categories: addition of more analysis features and provision for more flexible data storage and retrieval. Hopefully, these will gradually be added to the system in the future.

REFERENCES

1. Elkind, Jerome I., Starr, E. A., Green, D. M. and Darley, D. L., "Evaluation of a Technique for Determining Time-Invariant and Time-Variant Dynamic Characteristics of Human Pilots," NASA TN D-1897, May 1957.

2. Huggins, W. H., "Representation and Analysis of Signals, Part I: The Use of Orthogonalized Exponentials," AFCRC TR 57-357, Air Force Cambridge Research Center, Bedford, Mass., September 1957.

3. Elkind, J. I. and Green, D. M., "Measurement of Time-Varying and Nonlinear Dynamic Characteristics of Human Pilots," ASD Tech. Rept. 61-225, Wright-Patterson Air Force Base, Ohio, December 1961.

4. McQuillin, R. J., "DECAL-BBN Programming Manual," Bolt Beranek and Newman Inc Rpt. No. 1051, Cambridge, Mass., September 1963.

5. "Programmed Data Processor-1," Digital Equipment Corporation, Maynard, Massachusetts, 1961.